Bottom-up Reuse Guidelines: Metadata

Bottom-up Reuse Guidelines: Metadata and Its Relevance to Software Reuse

by Mary Hunter and Angelo Bertolli (Innovim / NASA GSFC)

Based on the presentation, "Metadata and Its Relevance to Software Reuse," by Angelo Bertolli and Mary Hunter, to the NASA Earth Science Data Systems Software Reuse Working Group, monthly telecon, December 20, 2006.

What Is Metadata?

Metadata is used to describe other data or software.

- Characterizing code
  > *Metadata for source code is used by version control systems to keep track of revisions and history. In this way source is treated as data that is being used by a version control software.*
- Descriptive information about the software
  > *Metadata also stores descriptive data about software or data. Some examples of metadata that humans (end users) would care about include the author or source, and brief and long descriptions. Some examples of metadata that other software applications might use include version numbers, timestamps, and identification strings.*
- Labeling, cataloging, and search information
  > *Metadata describes or defines other data to allow searching for a particular object. Often this metadata is stored in a database or a catalog, where the objects are labelled according to the desired search parameters.*

Why Use Metadata?

Metadata helps define the data in a concise way that helps different software use the same data in a consistent way. When used in application to resuable software, metadata is useful in two major areas: it can help categorize reusable assets, and following metadata standards facilitates reusing data by different software.

- Enables community to properly catalog and share software.
  > *When stored and distributed, software in either source or binary form is also data. It is useful to label, catalog, and search for different software titles, either as an end user or as a developer interested in software reuse.*
- Essential for explaining software assets.
  > *Metadata helps to describe reusable software assets by providing different levels of information about that asset. It can describe the programming language used, the compatible systems, and even dependencies.*
- Easier to find/search
  > *Users may search for and find software according to specific criteria such as operating system, stability level, or functional description.*
- Easier to reuse
  > *Metadata can make software easier to reuse at different levels by eliminating some of the investigation that needs to be done in the source code of the software. It can help a developer to understand more quickly how an asset should be integrated with an existing system.*
- Easier to modify
  > *Modifications can be kept track of and distributed. Metadata helps keep these modifications atomic so that consumers of the software select and use the components they want, while being able to create or modify other components.*

Metadata in Reuse

Metadata can be used to describe the data that needs to be manipulated by various pieces of software and can categorize software assets to allow developers to find something they can reuse. Using Metdata for source code promotes the future development of version control (VC) software because it allows newer VC software to understand the source code (data) of current systems. For example, metadata is used to convert CVS repositories to Subversion.

- Effective retrieval
- Systematic reuse
- Automatic routing based on status
- Tracking of reuse
- Reporting
- Contributes to a standard vocabulary for software reuse

Metadata Reuse Process

Since metadata is used to organize data, this makes individual objects easier to find. Reuse of software depends on discovering the reusable software components quickly and easily. With a large database of components, it would not be possible to read and know every component in detail in order to make a judgement on how best to reuse that component in your own software. This leads to a reusable asset repository which allows easy searching. The repository itself is really just a collection of metadata. Building a repository involves the following steps.

- Organizing software in a logical structure
- Categorizing software
- Create metadata catalog
- Populate with available software assets

Examples

- Insert metadata into software in the form of comments
  /* Input ASCII/TEXT Pass Plan */
  /* Output XML */
- Use standard software communication message
  - Standard must be defined prior to software creation
  - GMSEC

Using Metadata: del.icio.us

- Provides a repository of links that uses tags (keywords)
  > *del.icio.us uses metadata to allow users to share links to their favorite sites. Each user tags their link with keywords so that they or another user can later go and find links appropriate to a keyword or combination of keywords. Other metadata about the links that is presented is how many users have tagged a particular link with a particular keyword. This is considered to be a measure of popularity for a link.*
- Each user can freely select their own tags
  > *While some tags may be suggested when a new link is entered into the system, users are allowed to use any keyword they want. All tags are single-word identifiers, and the user separates them with spaces. If any tag doesn't already exist in the system, it is created by the first instance. Tags are case insensitive.*
- You can browse to a single tag to find things about that keyword
  Examples:
  - Find "software" sorted by popularity: http://del.icio.us/popular/software
  - Find popular things tagged with both "software" and "reuse": http://del.icio.us/popular/software+reuse

---

Using Metadata: GForge (example)

- Web based open source repository
  > *GForge uses data to categorize the resources in a software repository. When creating a new item, the creator selects from preset category trees, the categories under which their software falls.*
- Software is categorized in a heirarchy
  > *The system contains multiple category trees. Within each tree, a software item selects one specific node (instead of multiple nodes). Each leaf on a category tree implies that it is also of types of its ancestors.*
- Each resource is categorized
  > *Example: http://alioth.debian.org/projects/splashy/*
  > *Browse categories: http://alioth.debian.org/softwaremap/trove_list.php*

Tags vs Categories

Here is a brief summary of conceptual differences between tags and categories. Their concepts and usage may overlap, but they present a somewhat different environment.

• Tags provide meta information • Tags cross-connect (need to use multiple tags to be more specific) • Tags are usually chosen personally • Tags are usually a single word	• Categories organize resources heirarchically • Categories describe an exact point in the tree (cannot cross-connect) • Categories can have multiple words

---

References

1. CVS to Subversion with cvs2svn
2. Wikipedia pages
    - Metadata
    - Del.icio.us
    - Tags
    - Categorization
3. Tags vs Categories
    - Tags Are Not Categories
    - Categories versus Tags: What's the Difference and Which One?